Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of the claims in the application:

Listing of claims:

1-19. (Canceled)

20. (Currently Amended) A method, comprising:

responsive to an instruction pointer (IP) signal, that includes an IP tag and an IP set;

reading, from a plurality of ways, an entry tag, an entry bank number and entry data; ~~and~~

concurrently comparing (i) each entry tag and entry bank number with (ii) the IP tag and an identifier of each of the plurality of banks, wherein each entry is a way and the number of banks is independent from the number of ways; and

selecting data based on results of said comparing,

wherein the comparing includes:

concatenating (i) each entry tag and entry bank number;

concatenating (ii) the IP tag and a number representing each of the plurality of banks; and

comparing the concatenated entry tag and entry bank number with the concatenated IP tag and number representing each of the plurality of banks.

21. (Cancelled).

22. (Currently Amended) The method of claim ~~21~~20, wherein each entry further comprises an entry valid field and said selecting is further based on the entry valid field.

23. (Previously Presented) The method of claim 20, wherein the plurality of banks in the branch target buffer are implemented in a single array.

24. (Previously Presented) The method of claim 20, wherein each entry further comprises an entry valid field and different entries do not have identical entry tag, entry bank number and entry valid values.

25. (Previously Presented) The method of claim 20, further comprising decoding the IP set.

26. (Currently Amended) The method of claim ~~21~~20, wherein said selecting comprises selecting data to provide branch information related to a cache line to be read.

27. (Previously Presented) The method of claim 20, wherein each of the plurality of banks is organized in a set associative fashion.

28. (Previously Presented) A banked branch target buffer, comprising:

an input port to receive a look-up Instruction Pointer (IP) including an IP tag and an IP set, the IP set to identify a plurality of information entries including an entry tag, an entry bank number and entry data; and

a comparator coupled to said input port to concurrently compare (i) a concatenation of the IP tag and a bank identifier with (ii) a concatenation of the entry tag and entry bank number of each information entry, wherein each entry is a way and the number of banks is independent of the number of ways.

29. (Previously Presented) The banked branch target buffer of claim 28, wherein each entry further comprises an entry valid field and data is selected based on the entry valid field.

30. (Previously Presented) The banked branch target buffer of claim 28, wherein the branch target buffer has a plurality of banks implemented in a single array.

31. (Previously Presented) The banked branch target buffer of claim 30, wherein each of the plurality of banks is organized in a set associative fashion.

32. (Previously Presented) The banked branch target buffer of claim 28, wherein each entry further comprises an entry valid field and different entries do not have identical entry tag, entry bank number and entry valid values.

33. (Previously Presented) The banked branch target buffer of claim 28, wherein the IP set is decoded and used to read out the entries.

34. (Previously Presented) The banked branch target buffer of claim 28, wherein data is selected to provide branch information related to a cache line is to read.

35. (Previously Presented) A branch instruction prediction mechanism comprising:

a branch target buffer cache comprising a plurality of ordered branch target buffer banks formed as a single array, the array having a field to specify which bank an entry belongs to, each said ordered branch target buffer bank comprising a plurality of branch entries to store information about branch instructions addressed by address bits to specify a different subblock within said memory blocks;

a branch prediction circuit to receive said instruction pointer to index into all of said ordered branch target buffer banks of said branch target buffer cache, to concurrently compare (i) a concatenation of IP tags and an identifier of each of the plurality of ordered branch target buffer banks with (ii) a concatenation of entry tags and entry tag bank numbers, and to fetch at most one branch entry from each said plurality of branch target buffer banks based on said comparison; and

a prioritizer circuit to indicate the selection of one of said branch entries fetched by said branch prediction circuit from said ordered branch target buffer banks by selecting a first taken branch instruction located after said instruction pointer.

36. (Previously Presented) The branch instruction prediction mechanism of claim 35, wherein the branch prediction circuit is to predict a block of memory to fetch based on the instruction pointer that points to a currently executing instruction.

37. (Previously Presented) The branch instruction prediction mechanism of claim 36, wherein the branch instruction prediction mechanism is coupled to the block of memory.

38. (Previously Presented) The branch instruction prediction mechanism of claim 35, wherein the branch prediction circuit is to:

concatenate each entry tag with each entry bank number;

concatenate each IP tag with each identifier of each of the plurality of ordered branch target buffer banks; and

compare (i) the concatenated entry tags and entry bank numbers with (ii) the concatenated IP tags and identifiers of the plurality of ordered branch target buffer banks.

39. (Previously Presented) A method comprising:

receiving a current instruction pointer in a branch prediction mechanism, said branch prediction mechanism comprising a plurality of ordered branch target buffer banks formed as a single array, the array having a field to specify which bank an entry belongs to;

indexing into all of said plurality of ordered branch target buffer banks, each said branch target buffer bank comprising a plurality of branch entries to store information about branch instructions addressed by address bits to specify a different subblock within said memory blocks;

concurrently comparing (i) a concatenation of an IP tag of said instruction pointer and an identifier of each of the plurality of ordered branch target buffer banks with (ii) a concatenation of each entry tag and entry bank number of each one of said branch entries;

retrieving at most one ordered branch entry from each said ordered branch target buffer banks based on said comparison; and

selecting a next upcoming branch instruction from said retrieved ordered branch entries.

40. (Previously Presented) The method of claim 39, further comprising:

prior to the comparing,

concatenating the IP tag of said instruction pointer and the identifier of each of the plurality of ordered branch target buffer banks, and

concatenating each entry tag and entry bank number of each one of said branch entries.

41. (Previously Presented) The method of claim 39, further comprising:

predicting a subblock within the memory blocks to fetch based on the current instruction pointer that points to a currently executing instruction.

42. (Previously Presented) The method of claim 41, wherein the predicted subblock within the memory blocks includes the next upcoming branch instruction.

43. (Currently Amended) A method comprising:

receiving an Instruction Pointer (IP) signal, the instruction pointer signal including an IP tag and an IP set;

reading a plurality of entries corresponding to the IP set, each of the entries including an entry tag, an entry bank number, and entry data;

concatenating each entry tag with each corresponding entry bank number;

concatenating the IP tag with an identifier of each of the plurality of banks; ~~and~~

comparing the concatenated entry tags and corresponding entry bank numbers with the concatenated IP tag and identifiers; and

selecting data based on results of said comparing.

44. (Previously Presented) The method of claim 43, wherein the identifiers are constants.

45. (Previously Presented) The method of claim 43, wherein the comparing includes concurrently comparing the concatenated entry tags and corresponding entry bank numbers with the concatenated IP tag and identifiers.

46. (Currently Amended) The method of claim 45, wherein the comparing includes ~~optimally~~ sharing comparison logic to perform the concurrent comparing.

47. (Previously Presented) A cache, comprising:

an array of cache entries, the entries indexed by a set identifier and having outputs coupled to a plurality of output controllers, the array shared among a plurality of banks,

a plurality of bank indexing circuits, each bank indexing circuit comprising tag entries having fields for a tag identifier and a bank identifier corresponding to one of the plurality of banks, the tag entries indexed by a set identifier, and

a plurality of comparators, each comparator having a first input coupled to the tag entries and a second input coupled to an input identifier that includes an input tag identifier and a

respective bank identifier corresponding to one of the plurality of banks, each comparator having an output coupled to a respective output controller.

48. (Previously Presented) The cache of claim 47, wherein each tag entry is a concatenation of the tag identifier and the bank identifier.

49. (Previously Presented) The cache of claim 48, wherein the input identifier is a concatenation of the input tag identifier and the respective bank identifier.

50. (Previously Presented) The cache of claim 49, wherein each comparator is to compare a respective concatenated tag entry with a respective concatenated input identifier.

51. (Previously Presented) The cache of claim 50, wherein the comparators are to perform the comparisons concurrently.

52. (Previously Presented) The cache of claim 47, further comprising:

a selector to select one of the outputs from the comparators, the selection being a next upcoming branch instruction.

53. (Currently Amended) A method, comprising:

concurrently comparing (a) a plurality of tag entries, each tag entry comprising a concatenation of a tag identifier and a bank identifier with (b) a plurality of input identifiers, each input identifier comprising a concatenation of an input tag identifier and a respective bank identifier;

associating the bank identifier with one of a plurality of banks; and

sharing a single array of cache entries among the plurality of banks, each cache entry comprising branch instructions and each bank comprising a plurality of tag entries associated with corresponding cache entries.

54. (Cancelled).

55. (Previously Presented) The method of claim 54, further comprising:

selecting a cache entry based on the concurrent comparison to predict a next upcoming branch instruction.

56. (Cancelled).